Amendments to the Claims

1. (Previously presented) A computer implemented method of performing a transaction in a database system, comprising:

receiving a transaction to be performed, wherein the transaction is processed by a plurality of access modules; and

before any directive indicating commencement of an end transaction procedure is broadcast to the access modules, performing a flush of a transaction log from volatile storage to non-volatile storage by each of the access modules.

2. (Previously presented) The method of claim 1, further comprising issuing a request to flush the transaction log with a message sent to each of the access modules for performing a last step of the transaction, the last step performed prior to commencement of the end transaction procedure.

3. (Previously presented) The method of claim 2, further comprising performing the flush of the transaction log in a data access step prior to commencement of the end transaction procedure to avoid performance of a transaction log flush in the end transaction procedure.

4. (Previously presented) The method of claim 2, further comprising determining that the last step is being performed by all of the plurality of access modules involved in the transaction.

5. (Original) The method of claim 1, further comprising determining if the transaction log has been flushed before performing the end transaction procedure.

6. (Original) The method of claim 5, further comprising avoiding performance of a transaction log flush in the end transaction procedure if the transaction log has been flushed.

7. (Original) The method of claim 1, further comprising:
identifying the transaction as an implicit transaction.

8. (Previously presented) The method of claim 1, further comprising:
performing the end transaction procedure.

9. (Previously presented) The method of claim 8, performing the end transaction procedure comprising:
skipping broadcast of the directive indicating commencement of the end transaction procedure to the plurality of access modules.

10. (Previously presented) A computer implemented method of performing an end transaction procedure in a database system, comprising:
after commitment of a transaction, a first access module in the database system writing an end transaction indication to a first transaction log portion in volatile storage, the first access module being part of a cluster of access modules; and
the first access module sending an end transaction directive to a fallback access module associated with the first access module, the fallback access module being part of the cluster.

11. (Previously presented) The method of claim 10, wherein the first access module sends the end transaction directive to the fallback access module but not to other access modules in the cluster.

12. (Original) The method of claim 10, wherein sending the end transaction directive comprises sending an end transaction-part one directive.

13. (Previously presented) The method of claim 12, further comprising the fallback access module broadcasting an end transaction-part two directive to all access modules in the cluster.

14. (Previously presented) The method of claim 10, further comprising the fallback access module writing an end transaction indication to a second transaction log portion in volatile storage.

15. (Previously presented) The method of claim 10, further comprising the first access module flushing the first transaction log portion from volatile storage to non-volatile storage.

16. (Previously presented) The method of claim 10, further comprising the first access module flushing the first transaction log portion from volatile storage to non-volatile storage but the other access modules in the cluster not flushing their respective transaction log portions.

17. (Previously presented) A database system comprising:

persistent storage;

volatile storage; and

a plurality of access modules, wherein each access module is coupled to the persistent storage and the volatile storage; and

each of the access modules being adapted to flush a transaction log maintained by the access module from the volatile storage to the persistent storage before any directive indicating commencement of an end transaction procedure is broadcast to the access modules.

18. (Previously presented) The database system of claim 17, further comprising a controller adapted to determine if each access module has flushed the transaction log maintained by the access module before commencement of the end transaction procedure.

19. (Previously presented) The database system of claim 18, wherein the controller is adapted to skip sending a directive to perform a transaction log flush if the

controller determines that each access module has flushed the transaction log before commencement of the end transaction procedure.

20. (Previously presented) The database system of claim 17, further comprising a controller adapted to provide a flush directive with a message to each of the access modules to perform a last step of the transaction before commencement of the end transaction procedure.

21. (Previously presented) An article comprising a computer readable storage medium storing instructions for enabling a processor-based system to:

receive a transaction to be performed, wherein the transaction is processed by a plurality of access modules;

determine that a last step of the transaction involves the plurality of access modules, wherein the last step is performed before any directive indicating commencement of an end transaction procedure is broadcast to the access modules; and

flush a transaction log from volatile storage to a non-volatile storage while the last step is performed by the plurality of access modules.

22. (Previously presented) The article of claim 21, further storing instructions for enabling the processor-based system to:

perform the end transaction procedure, wherein the end transaction procedure follows execution of the last step of the transaction.

23. (Previously presented) The article of claim 22, further storing instructions for enabling the processor-based system to:

avoid broadcast of any directive indicating commencement of the end transaction procedure to the plurality of access modules.

24. (Previously presented) A computer implemented method of performing a transaction in a database system, comprising:

receiving a transaction to be performed on plural access modules in the database system;

maintaining a log in volatile storage to track operations performed in the transaction; and

writing the log to persistent storage before any directive indicating commencement of an end transaction procedure is broadcast to the plural access modules.

25. (Original) The method of claim 24, wherein writing the log to persistent storage comprises flushing the log.

26. (Original) The method of claim 24, wherein maintaining the log comprises maintaining a transaction log.

27. (Original) The method of claim 24, further comprising performing the end transaction procedure, the end transaction procedure comprising writing an end transaction indication into the log.

28. (Previously presented) A database system comprising:

persistent storage;

volatile storage;

access modules coupled to the persistent storage and the volatile storage; and

a parsing engine coupled to the access modules, the parsing engine adapted to perform one of:

(a) providing a directive with a message to perform a last step of a transaction and communicating the directive to the access modules, each access module responsive to the directive to perform a transaction log flush from the volatile storage to the persistent storage before any directive indicating commencement of an end transaction procedure is broadcast to the access modules; and

(b) determining if each of the access modules has performed a transaction log flush before start of the end transaction procedure;

the parsing engine adapted to avoid sending a broadcast directive to the access modules to cause performance of a transaction log flush during the end transaction procedure.

29. (Previously presented) The method of claim 1, wherein the transaction comprises plural steps, the method further comprising:

performing the plural steps prior to performing the end transaction procedure, and

wherein performing the flush of the transaction log comprises performing the flush of the transaction log in one of the plural steps.

30. (Previously presented) The method of claim 29, wherein performing the plural steps comprises performing, in each of the plural steps, access of relational table data stored in the database system.

31. (Previously presented) The method of claim 29, wherein performing the flush of the transaction log in one of the plural steps comprises performing the flush of the transaction log in a last one of the plural steps.

32. (Previously presented) The method of claim 1, further comprising each access module adding a first entry to the transaction log to redo the transaction by the access module in case of system failure.

33. (Previously presented) The method of claim 4, wherein performing the flush of the transaction log is prior to commencement of the end transaction procedure if the last step is performed by all of the plurality of access modules, the method further comprising:

performing the flush of the transaction log in the end transaction procedure if the last step is not performed by all of the plurality of access modules.

34. (Previously presented) The database system of claim 17, wherein the access modules are further adapted to perform a transaction comprising plural steps, and to perform the flush of the transaction log in one of the plural steps.

35. (Previously presented) The database system of claim 34, wherein the one of the plural steps comprises a last one of the steps.

36. (Previously presented) The database system of claim 35, wherein the transaction log comprises a first entry associated with each access module to enable a redo of the transaction in case of system failure.

37. (Previously presented) The database system of claim 36, wherein the transaction log further comprises a second entry associated with each access module to enable an undo of the transaction.

38. (Previously presented) The database system of claim 34, further comprising a controller adapted to determine whether a last one of the steps involves all the access modules, and in response to determining that the last one of the steps involves all the access modules, the controller further adapted to send a directive to all the access modules to perform the flush of the transaction log in the last one of the steps.

39. (Previously presented) The database system of claim 38, in response to determining that the last step does not involve all access modules, the controller further adapted to send a directive to perform the flush of the transaction log in the end transaction procedure.

40. (Previously presented) The article of claim 21, wherein the transaction comprises plural steps, the article further storing instructions for enabling the processor-based system to:

perform the plural steps prior to commencement of the end transaction procedure, and

wherein performing the flush of the transaction log comprises performing the flush of the transaction log in one of the plural steps.

41. (Previously presented) The article of claim 40, wherein performing the plural steps comprises performing, in each of the plural steps, access of relational table data stored in a database system.

42. (Previously presented) The article of claim 40, wherein performing the flush of the transaction log in one of the plural steps comprises performing the flush of the transaction log in a last one of the plural steps.

43. (Previously presented) The article of claim 42, further storing instructions for enabling the processor-based system to cause each access module to add a first entry to the transaction log to redo the transaction by the access module in case of system failure.